Lecture 14: Fast algorithms for robust mean estimation via MMW

November 12, 2025

The goal of this lecture is to sketch out the basic ideas for how to use MMW to obtain nearly-linear time algorithms for robust mean estimation. This will follow the presentation in [1], and an interested reader is encouraged to look there for a more complete writeup of this algorithm.

1 Recall: matrix multiplicative weight update

We first briefly recall the matrix learning from experts game, and the MMW update rule. We will actually use the following slight specification of it:

Theorem 1.1 ([2]). *Let $G^{(1)}, \ldots, G^{(T)}$ be a sequence of positive semi-definite gain matrices, and let α be chosen so that $\alpha G^{(t)} \preceq I$ for all iterations $t = 1, \ldots, T$. Then, define*

$$P^{(t)} = \frac{\exp\left(\alpha \sum_{i=1}^t G^{(t)}\right)}{\operatorname{tr}\exp\left(\alpha \sum_{i=1}^t G^{(t)}\right)} \; . \tag{1}$$

Then, we have that

$$\left\| \sum_{t=1}^T G^{(t)} \right\|_\infty \leq \sum_{i=1}^T \langle G^{(t)}, P^{(t)} \rangle + \alpha \sum_{t=1}^T \left\langle P^{(t)}, G^{(t)} \right\rangle \cdot \|G^{(t)}\|_\infty + \frac{\log n}{\alpha} \; .$$

Note that the step-size was denoted ε last lecture, but we will use α to avoid conflicting with ε-corruption.

2 Iteration complexity of filtering

We recall the fundamental geometric problem underlying robust mean estimation. We are given a dataset $S = S_{\mathrm{good}} \cup S_{\mathrm{bad}} \setminus S_r$ of n points in $\mathbb{R}^d$ so that (i) we have that S_{good} is a set of n points with mean μ and

$$\Sigma_g = \frac{1}{n} \sum_{i \in S_{\mathrm{good}}} (X_i - \mu)(X_i - \mu)^\top \preceq I \; ,$$

and (2) we have that $|S_{\mathrm{bad}}| = |S_r| = \varepsilon n$. Now, the objective of filtering is to certify that the empirical covariance $\widehat{\Sigma}$ of S has spectral norm which is at most $O(1)$. Recall that the key algorithmic idea is that if the spectral norm of the empirical covariance is large, then there is a univariate filter: namely, we define scores

$$\tau_i = \langle v, X_i - \widehat{\mu} \rangle^2 \; ,$$

where v is the top eigenvector of $\widehat{\Sigma}$, and $\widehat{\mu}$ is the empirical mean, and then we downweight points based on how large these scores are. For simplicity, let's pretend there is an idealized filter, which given non-negative scores τ_i satisfying

$$\sum_{i \in S_{\mathrm{good}} \setminus S_r} \tau_i \leq \frac{1}{4} \sum_{i \in S_{\mathrm{bad}}} \tau_i \; , \tag{2}$$

always deterministically removes more bad points than good points.[1] Such a filter does not exist, so to make the full argument formal, one would actually either need to randomize, or use a randomized filter.

Fast filtering in 1D Right now, even in 1 dimension, it's not so clear why filtering is fast: indeed, the "simple" analysis only gives that the filter would require εn iterations, even if the scores were kept the same in every iteration, for the filter to terminate, since our progress condition is that we remove at least one point every iteration.

However, this is a bit naive, and it turns out one can do something slightly better:

Lemma 2.1. *There is an (randomized) algorithm, which given τ_i satisfying* (2)*, outputs a set of points S' so that with high probability, we've removed more bad points than good points, and $\sum_{i \in S'} \tau_i \leq \frac{1}{3} \sum_{i \in S} \tau_i$. Moreover, this algorithm runs in nearly linear time.*

We leave the details of this algorithm as an exercise to the reader (i.e., homework). In the remainder of this section, once again, let's pretend there is no randomness involved, and that this algorithm can be made deterministic.

Barriers for filtering in high dimensions However, even a fast 1D filter is not necessarily useful, as it's not clear how to convert its guarantees to the high-dimensional setting. Previously, we argued briefly that a simple argument guarantees that the filter will only have to run for at most $O(d)$ iterations, yielding an overall runtime for the filter algorithm of $O(nd^2)$, since each iteration can be performed in nearly-linear time. However, it's also straightforward to see that such a time complexity is inevitable for the basic filter algorithm. Consider the following instance: let S_{good} be a set of $(1-\varepsilon)n$ points in isotropic position: that is, its mean is zero and the covariance of S_{good} is I, for $n \gg d/\varepsilon$. Consider the following choice of S_{bad}: let e_i denote the i-th standard basis vector, and let S_{bad} put $\varepsilon n/(2d)$ points at $10i\sqrt{d}/\sqrt{\varepsilon} \cdot e_i$, and $\varepsilon n/(2d)$ points at $-10i\sqrt{d}/\sqrt{\varepsilon} \cdot e_i$. Then, the empirical mean of the corrupted dataset is actually correct: it is 0, but the algorithm will not know this. The empirical covariance of the corrupted dataset will be as follows:

$$(1-\varepsilon)I + \sum_{i=1}^{d} 100 i^2 e_i e_i^\top \; .$$

We can easily read off the top eigenvector from this: namely, it is e_d. When we filter, we will then clearly remove all of the points in S_{bad} along this direction, but none of the others. As a result, we will have to repeat this process once for every standard basis vector! Thus, one can clearly see that this must repeat d times.

The problem here is that it's very inefficient to filter along every direction separately, at least in this instance. Rather, in this case, it's actually really clear which points are bad: in particular, all of the bad points have ℓ_2 norm significantly larger than $\sqrt{d}/\varepsilon$. In contrast,

$$\frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} \|X_i\|_2^2 = \operatorname{tr}\left(\frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i X_i^\top \right) = \operatorname{tr}(I) = d \; ,$$

so on average most good points have ℓ_2 norm $\sqrt{d}$. Indeed, by Chebyshev's inequality, in this case one can show that filtering based on the score $\tau_i = \|X_i\|_2^2$ actually would work. However, this clearly also doesn't work in general. For instance, if all of the bad points are equal to $10/\sqrt{\varepsilon} \cdot e_i$, then this still induces a large eigenvalue, but these points have constant norm (at least when ε is small).

The key geometric insight is that there is a clean way of interpolating between these two extremes: intuitively, when there is only one bad direction, then it suffices to filter using just a single eigenvalue. But

[1] Note that previously we only used the weaker condition that $\sum_{i \in S_{\text{good}} \setminus S_r} \tau_i \leq \sum_{i \in S_{\text{bad}}} \tau_i$, however, the choice of constant there was arbitrary, and we could make the sum of the scores of the good points smaller by any constant amount we want.

when there are many bad directions, this means that the bad points in those directions are correspondingly larger, and so we can still filter many of them all at once.

What this really suggests is to find a certificate that the spectral norm of the covariance is large which witnesses as many large directions as possible. One way to do this in a principled fashion is as follows. Recall that the spectral norm is dual to the Schatten-1 norm, which amongst other things, implies that for all M PSD, we have that

$$\|M\|_\infty = \sup_{U \succeq 0 : \mathrm{tr}(U) = 1} \langle M, U \rangle \; .$$

Now, this is optimized by $U = vv^\top$ for v being the top eigenvector of M, but for our purposes, it suffices to find *any* valid certificate that witnesses that $\langle U, M \rangle \geq \Omega(1)$. In fact, by the exact same logic as before, if we can find such an U, then if we let

$$\tau_i = (X_i - \widehat{\mu})^\top U (X_i - \widehat{\mu}) \; ,$$

then filtering with these scores will suffice. So this motivates us to find a certificate U which witnesses as many large eigenvectors of M as possible. One clean way to do this is to find U which maximizes the following objective:

$$\max \alpha \cdot \langle U, \widehat{\Sigma} \rangle + S(U) \quad \text{such that} \quad U \succeq 0, \mathrm{tr}(U) = 1 \; .$$

Here $S(U)$ is the *von Neumann* entropy of U, which is the entropy of the eigenvalues of U: that is, if $lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$, then

$$S(U) = \sum \lambda_i \log(1/\lambda_i) \; .$$

Here is the intuition: because the eigenvalues of U are nonnegative and sum to 1, they specify a distribution over the eigenvectors. Then, $S(U)$ is the entropy of this distribution. So this objective is trying to find a maximally spread certificate that the spectral norm is large. The point is that the maximizer of this objective actually has a nice closed form: in fact, it is exactly

$$U = \frac{\exp\left(\alpha \widehat{\Sigma}\right)}{\mathrm{tr} \exp\left(\alpha \widehat{\Sigma}\right)} \; , \tag{3}$$

which looks like the MMW update! This suggests that perhaps we should consider using a MMW-style update for this problem.

3 Fast filtering via MMW

Consider the following slightly easier problem: we have a set of points S with mean $\widehat{\mu} = 0$ and empirical covariance $\widehat{\Sigma}$ satisfying $\|\widehat{\Sigma}\|_\infty = C$, and our goal is to filter away points so that the resulting covariance has spectral norm at most $C/2$. If we can solve this simpler problem, its not hard to see that by repeatedly applying this primitive to the centered set of points, we can, in at most logarithmically many iterations, decrease the spectral norm of the overall dataset down to $O(1)$.

Here is the basic idea: we will play the following matrix learning from experts game as the adversary. For any set of points T, we let $M(T)$ denote empirical second moment of the points in T. When $T = S$, this is the empirical covariance, since $\widehat{\mu} = 0$. However, for other sets, this may not be the case.

We maintain a set of points S_t, initially set to $S_1 = S$. For $t = 1, \ldots, T$, we do the following:

- First, if $\|M(S_t)\|_\infty < C/2$, we terminate.

- Otherwise, the MMW will commit to a density matrix $P^{(t)}$, which we will interpret as MMW's attempt at finding a good certificate that $\Sigma(S_t)$ has large spectral norm.
- We will then compute $C_t = \langle M(S_t), P^{(t)} \rangle$.
- If this is less than $C/4$, we do nothing, and set $S_{t+1} = S_t$.
- Else, we define scores
$$\tau_i = (X_i - \mu(S_t)) P^{(t)} (X_i - \mu(S_t)) \; , \tag{4}$$
and we let S_{t+1} be the result of filtering (using Lemma 2.1) with these scores.
- We then feed in the gain matrix $G^{(t)} = M(S_{t+1})$ into the MMW algorithm.

Notice that by doing this, MMW's gain matrices will have the form

$$P^{(t)} = \frac{\exp\left(\alpha \sum_{t=1}^{T} M(S_t)\right)}{\operatorname{tr}\exp\left(\alpha \sum_{t=1}^{T} M(S_t)\right)} \; ,$$

which is essentially the form of the solution in (3), but averaged across many iterations.

We now prove correctness of this algorithm. First, we wish to argue that we never remove more good points than bad points. We will just sketch the proof here: again, the key idea is that because

$$\frac{1}{|S_t|} \sum_i \tau_i = \langle M(S_t), P^{(t)} \rangle \geq C/4 \; ,$$

then since $P^{(t)} \succeq 0$ and has trace 1, this is a valid certificate that the spectral norm of $M(S_t)$ is at least C_t! Meanwhile, the score of the good points cannot be large, because they still have bounded covariance, and hence $\langle M, P^{(t)} \rangle = O(1)$. One needs to worry about the fact that we are not centering at the mean of the good points, but again, this is fine because of the same analysis as before, by leveraging the spectral signatures lemma.

So it remains to understand what the iteration complexity of this algorithm is. For this, let's plug in the guarantee of MMW with these matrices: by the previous lecture, the guarantee is that, for any appropriate choice of step-size α, we have that

$$\left\| \sum_{t=1}^{T} M(S_{t+1}) \right\|_\infty \leq \sum_{i=1}^{T} \langle M(S(t+1)), P^{(t)} \rangle + \alpha \sum_{t=1}^{T} \langle M(S(t+1)), P^{(t)} \rangle \cdot \|M(S_{t+1})\|_\infty + \frac{\log n}{\alpha} \; .$$

First, observe that by our safety condition, S_t is always a set of at least $(1-2\varepsilon)n$ points, since we never throw out more bad points than good points. From this, we obtain the following naive bound:

$$\|M(S_t)\|_\infty \leq \frac{1}{1-2\varepsilon} \|M(S)\|_\infty \leq \frac{C}{1-2\varepsilon} \; .$$

One particular implication of this is that we can always take $\alpha = \frac{1-2\varepsilon}{C}$, and this will satisfy the constraints of the theorem.

But here's the somewhat magical thing: if we don't choose to filter in iteration t, then $\langle M(S(t+1)), P^{(t)} \rangle \leq C/4$ by definition. Otherwise, recall that by Lemma 2.1, it holds that

$$\langle M(S(t+1)), P^{(t)} \rangle \leq \frac{1}{3} \langle M(S(t)), P^{(t)} \rangle \leq \frac{1}{3} \|M(S_t)\|_\infty \leq \frac{1}{3(1-2\varepsilon)} C \leq \frac{2}{5} C \; ,$$

for ε sufficiently small. Thus no matter if we choose to filter or not in iteration t, we have the guarantee that $\langle M(S(t+1)), P^{(t)} \rangle \leq \frac{2}{5} C$. So, plugging this in, we obtain that the RHS is upper bounded by

$$\frac{2}{5} \cdot CT + \alpha \frac{2}{5(1-2\varepsilon)} \cdot C^2 T + \frac{\log n}{\alpha} \; .$$

On the other hand, we observe that for all $t < T+1$, we have:

$$\begin{aligned} M(S_{T+1}) &= \frac{1}{|S_{T+1}|} \sum_{i \in S_{T+1}} X_i X_i^\top \\ &\preceq \frac{1}{|S_{T+1}|} \sum_{i \in S_t} X_i X_i^\top \\ &\preceq M(S_t) \; . \end{aligned}$$

Importantly, this implies that

$$\left\| \sum_{t=1}^{T} M(S_{t+1}) \right\|_\infty \geq T \cdot \|M(S_{T+1})\|_\infty \; .$$

Putting these guarantees together, we obtain:

$$T \cdot \|M(S_{T+1})\|_\infty \leq \frac{2}{5} \cdot CT + \alpha \frac{2}{5(1-2\varepsilon)} \cdot C^2 T + \frac{\log n}{\alpha} \; .$$

Setting $\alpha = \frac{1-2\varepsilon}{C}$, and dividing through by T, we obtain that

$$\|M(S_{T+1})\|_\infty \leq \frac{2}{5} \cdot C + \frac{2}{5} \cdot C + \frac{1}{T} \cdot \frac{1}{1-2\varepsilon} C \log n \; .$$

In particular, if we take $T = O(\log n)$, this implies that $\|M(S_{T+1})\|_\infty \leq C/2$. Thus, this algorithm halves the spectral norm in logarithmically many iterations!

It is a natural question whether or not it is actually necessary to do this "epoch"-based approach, where one halves the spectral norm in every epoch of the algorithm. It turns out this is not possible; we leave it as an exercise to the reader to do this properly.

Implementing MMW efficiently There is a final detail, which I will mostly skim over, which is how to actually implement each iteration of MMW efficiently. As before, computing the spectral norm efficiently can be done via the power method in nearly linear time. The main bottleneck is the computation of the $P^{(t)}$ matrices. In fact, we cannot, in general, compute these matrices! This is because to compute a matrix exponential requires taking an SVD, which requires super-linear runtime.

The key observation is that we do not actually need a full representation of the $P^{(t)}$ matrices. Rather, we only need to be able to form the scores τ_i, which consist of matrix-vector products with the $P^{(t)}$ matrices. For this, it turns out we can leverage standard ideas from the literature on fast numerical linear algebra: we defer the reader to [1] for a more complete description of these tricks.

References

[1] Yihe Dong, Samuel B Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. *arXiv preprint arXiv:1906.11366*, 2019.

[2] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 237–245. ACM, 2015.